

Jarno Hujanen

## KELLONAJAN JAKOLAITE

Insinöörityö  
Kajaanin ammattikorkeakoulu  
Tekniikan ja liikenteen ala  
Tietotekniikan koulutusohjelma  
Kevät 2004

Osasto Tekniikka ja liikenne	Koulutusohjelma Tietotekniikka
Tekijä(t) Jarno Hujanen	
Työn nimi Kellonajan jakolaite	
Vaihtoehtoiset ammattiopinnot Langaton tiedonsiirto	Ohjaaja(t) Jukka Heino
Aika 31.3.2004	Sivumäärä 44 + 16
<p>Tiivistelmä</p> <p>Insinööritöiden tavoitteena oli suunnitella ja toteuttaa laite, joka lähettää kellonajan ja päivämäärän Kajaanin ammattikorkeakoulun tietoverkkoon minuutin välein. Lisäksi vaatimuksena oli kehittää PC-tietokoneelle ohjelma, joka vastaanottaa laitteen lähettämät tiedot.</p> <p>Laite toteutettiin Intel 8051 -prosessorilla sekä Moxa-merkkisellä sarjaliikennemuuntimella. Prosessorille laadittu ohjelma toteutettiin käyttäen IAR-kehitysympäristöä sekä prosessorille tarkoitettua emulaattoria. Ohjelmointikielenä käytettiin C-kieltä. PC-tietokoneelle tehtiin ohjelma, joka vastaanottaa laitteen lähettämät tiedot ja muuttaa kellonajan ja päivämäärän niitä vastaaviksi. PC-ohjelma toteutettiin C++-ohjelmointikielillä käyttäen apuna WinSock-rajapintaa.</p> <p>Kellonajan jakolaitteen toiminta testattiin Kajaanin ammattikorkeakoulun laboratoriossa ja tulokset olivat odotusten mukaiset. Laitteen käynti 24 tunnin aikana oli alle yhden sekunnin, joten sen voitiin todeta olevan varsin tarkka. PC-työaseman ohjelman toimivuus todettiin laitteen testauksen yhteydessä.</p> <p>Kellonajan jakolaitteelle on monia jatkokehitysmahdollisuuksia, joista tärkeimmät ovat piirilevyn suunnittelu ja koteloinnin rakentaminen. PC-tietokoneen ohjelma ei sovellu ajettavaksi Windows-ympäristössä tällaisenaan. Pienellä kehittämisellä sen saisi sulautettua Windowsin käyttöliittymään.</p>	
<p>Luottamuksellinen</p> <p>Kyllä</p> <p>Ei                      X</p>	
<p>Hakusanat</p> <p>Kellonajan jakolaite, Intel 8051, sarjaportti, RS-232, sarjaliikennemuunnin, WinSock</p>	
<p>Säilytyspaikka</p> <p>Kajaanin ammattikorkeakoulun kirjasto</p>	

Faculty Faculty of Engineering	Degree programme Information Technology
Author(s) Jarno Hujanen	
Title A Time Sharing Device	
Optional professional studies Wireless Communication	Instructor(s) / Supervisor(s) Jukka Heino
Date 31 March 2004	Total number of pages 44 + 16
<p>Abstract</p> <p>The purpose of this final year project was to design and build a device which transmits the time and date to Kajaani Polytechnic's network. There was also a requirement to develop a PC program which receives the data sent by the time sharing device.</p> <p>The time sharing device is based on the Intel 8051 processor and Moxa-serial converter. The software for the processor was made by using the IAR development environment and 8051 emulator. The programming language was ANSI C. The PC program was developed to receive the time and date from the time sharing device. It updates the computer's time and date every minute. The program was made by the programming language C++ and is based on the WinSock.</p> <p>The time sharing device was proved to work in Kajaani Polytechnic's laboratories. Within 24 hours the internal clock of the device altered less than a second compared to a real time clock. The device was discovered to be quite exact. The PC program was also proved to work as it was supposed to.</p> <p>There are a plenty of possibilities to improve the function of the time sharing device. The most important details to improve are to design a printed circuit board and to build a covering. The PC program like this is not suitable for the Windows environment. By developing the program it could be integrated into the Windows user interface.</p>	
<p>Confidential</p> <p>Yes</p> <p>No            X</p>	
<p>Keywords</p> <p>Time sharing device, Intel 8051, serial port, RS-232, serial converter, WinSock</p>	
<p>Deposited at</p> <p>Kajaani Polytechnic's library</p>	

## ALKUSANAT

Tämä insinöörityö on tehty Kajaanin ammattikorkeakoululle. Insinöörityön tekeminen on ollut mielenkiintoista ja haastavaa. Haluan kiittää valvojaani, diplomi-insinööri Jukka Heinoa työn valvonnasta, ohjauksesta sekä mielenkiintoisesta aiheesta. Lisäksi haluan kiittää ammattikorkeakoulun opettajia Eero Soinista ja Kaisu Korhosta työn kielellisestä ohjauksesta. Erityiskiitoksen ansaitsee kehitys-insinööri Markku Karppinen, joka on hoitanut insinöörityöni tarviketilaukset.

Kajaanissa 31.3.2004

Jarno Hujanen

## SISÄLTÖ

1	JOHDANTO .....	6
2	YLEISTÄ .....	7
2.1	Ajanlasku .....	7
2.2	Ajanmittaus .....	8
2.3	Kello .....	8
2.4	PC:n kello .....	9
2.5	Navigointi .....	10
3	KYTKENTÄ .....	12
3.1	Laitteelle asetetut vaatimukset .....	12
3.2	Proessori .....	13
3.3	Muistipiiri .....	14
3.4	Sarjaliikenne .....	15
4	SARJALIIKENNEMUUNNIN .....	18
5	OHJELMA PROSESSORILLE .....	20
5.1	Sarjaportti .....	20
5.2	Toiminta .....	21
5.3	Muistipiiri .....	21
6	OHJELMA TYÖASEMALLE .....	22
6.1	Protokollat .....	22
6.2	Socket .....	23
6.3	Toiminta .....	23
7	TESTAAMINEN .....	24
8	JATKOKEHITYS .....	25
9	YHTEENVETO .....	26
	LÄHDELUETTELO .....	27
	LIITTEET	

## 1 JOHDANTO

Tämän insinöörityön tavoitteena oli rakentaa laitteisto, joka jakaa kellonaikaa Kajaanin ammattikorkeakoulun tietoverkkoon. Työ on tehty ammattikorkeakoulun tilauksesta.

Valitsin insinöörityöni aiheeksi kellonajan jakolaitteen, koska tietoverkot ja laiteläheinen ohjelmointi herättivät kiinnostukseni koulutuksen aikana. Laitteen on tarkoitus toimia kellonajan jakolaitteen prototyyppinä, ja sille on monia jatkokehitysmahdollisuuksia. Ammattikorkeakoulun on tarkoitus ottaa laite käyttöön tulevaisuudessa.

Työ toteutettiin käyttämällä Intel 8051 -prosessoria ja Moxa-sarjaliikennemuunninta. Prosessorille keskeytyksiä aiheuttaa GPS-laite, joka on kytketty prosessorin ulkoiseen keskeytykseen. Prosessorille tehty ohjelma laskee tulleita keskeytyksiä ja päivittää niiden avulla kelloaan ja kalenteriaan. Laite lähettää tiedot minuutin välein sarjaliikennemuuntimelle, joka lähettää ne edelleen tietoverkkoon. PC-työasemaan tehtiin ohjelma, joka vastaanottaa tiedot.

Työn toisessa luvussa perehdytään ajanlaskuun ja -mittaukseen sekä kelloihin ja navigointiin. Neljässä seuraavassa luvussa perehdytään laitteen suunnitteluun ja toteuttamiseen. Seitsemännessä luvussa laite testataan ja kahdeksannessa pohditaan jatkokehitysmahdollisuuksia. Työn loppuun on liitetty työn kannalta oleellisia dokumentteja, kuten ohjelmalistauksia, kytkentäkaavio, ohjelmien toiminnallisia vuokaavioita sekä käyttöohje. Tässä työssä ei käsitellä perusasioita, koska työn lukijan oletetaan olevan tekniikan alan ammattilainen tai muuten asiaan perehtynyt.

## 2 YLEISTÄ

Ajan laskeminen ja mittaaminen on kiinnostanut ihmisiä jo vuosituhansien ajan. Käsiteltäessä pitkiä ajanjaksoja puhutaan yleensä ajanlaskusta. Kun kyseessä ovat alle vuorokauden mittaiset jaksot, puhutaan ajanmittaamisesta. Perustan ajanlaskulle tarjoaa luonnonilmiöiden lähes säännöllinen toistuvuus.

### 2.1 Ajanlasku

Suomessa on käytössä gregoriaaninen kalenteri, jonka paavi Gregorius XIII otti käyttöön vuonna 1582 [1, s. 533]. Ennen gregoriaanisen kalenterin käyttöönottoa Suomessa oli käytössä juliaaninen kalenteri ja sen paranneltu versio. Suomessa gregoriaanista kalenteria on käytetty vuodesta 1869 [2, s. 46]. Kalenteri on tapa määritellä vuoden, kuukauden, viikon ja päivän pituus. Vuodella tarkoitetaan vuodenaikojen toistumisjaksoa, joka tapahtuu maapallon kiertäessä Aurion ympärillä. Perustana ajanlaskulle toimii trooppinen vuosi, jonka pituus on 365 vuorokautta, 5 tuntia, 48 minuuttia ja 46 sekuntia kevättasauspisteen suhteen [1, s. 2169]. Kuukaudella tarkoitettiin alun perin aikaa uuden kuunsirpin ilmestymisestä uuteen kuuhun [2, s. 7]. Viikko on seitsemän vuorokauden jakso. Vuoden ensimmäiseksi viikoksi merkitään se viikko, joka sisältää vuoden ensimmäisen torstaipäivän. [2, s. 18]. Kalentereissa päivällä tarkoitetaan vuorokautta, jonka pituus on 24 tuntia.

Gregoriaanisen kalenterin mukaan joka neljäs vuosi on karkausvuosi, jossa on 366 vuorokautta. Poikkeuksena ovat täydet vuosisadat, jotka eivät yleensä ole karkausvuosia, lukuun ottamatta vuosia, jotka ovat jaollisia 400:lla [2, s. 14]. Tällöin voidaan sanoa tällä vuosisadalla joka neljännen vuoden olevan karkausvuosi. Vuoteen 2000 asti karkauspäivä merkittiin 23. helmikuuta jälkeiseksi päiväksi, mutta vuodesta 2000 alkaen karkauspäivä on ollut 29. helmikuuta.

## 2.2 Ajanmittaus

Koska vuosi jaettiin 12:een osaan, oli luontevaa jakaa myös vuorokausi 12:een päivä- ja yötuntiin. Vuorokausi koostuu 12:sta päivä- ja yötunnista, jolloin vuorokauden pituudeksi saadaan 24 tuntia. Tunti jaettiin 60:een osaan, jolloin tunti on 60 minuuttia. Jakajana toimiva luku 60 on babylonialaisten numerojärjestelmän kantaluku [2, s. 77]. Sekunteihin päästiin jakamalla minuutti 60:een osaan, jolloin minuutissa on 60 sekuntia. Sekunti on SI-järjestelmässä (Système Internationale) ajan perusyksikkö, ja sen pituus määritellään cesium-atomin värähtelyjen avulla [3, s. 37]. Aikaan tehdään korjauksia maan pyörimisnopeuden muutosten takia karkaussekunnin avulla [1, s. 817]. Karkaussekunti lisätään aikaan yleensä noin 1–2 vuoden välein, ja niitä voidaan lisätä kesä- tai joulukuun lopussa.

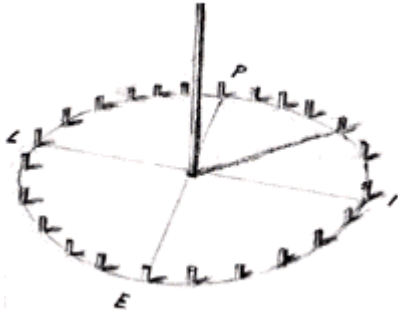
Maapallo on jaettu 24:een aikavyöhykkeeseen, joista jokainen on 15 astetta leveä. Kuljettaessa vyöhykkeeltä toiselle ajan muutos on yksi tunti, riippuen kulusuunnasta. Nollameridiaaniksi on sovittu Greenwichin meridiaani, joka sijaitsee Englannissa, Lontoon lähellä. Tästä johtuen Suomessa aika on kaksi tuntia edellä Greenwichin aikaa (GMT). Nykyisin kellomme käyvät niin sanottua koordinoitua yleisaikaa (UTC). Nimenomaan UTC-ajassa käytetään edellä mainittuja karkaussekunteja.

## 2.3 Kello

Kello on laite, joka mittaa ajan kulumista. Vanhimmat kellot olivat vesi-, hiekka-, aurinko- ja tulikelloja. Aurinkokellona toimii keppi, joka pystytetään maahan. Kepin ympärille piirretään ympyrä, joka jaetaan 24:ään yhtä suureen osaan. Aurinkokellon toiminta ilmenee kuvasta 1. Vesikello perustuu veden valumiseen pienen reiän kautta, jolloin kellonaika voidaan todeta säiliössä olevan veden tasosta. Hiekkakello tunnetaan nykyisin paremmin nimellä tiimalasi, ja sen toiminta perustuu hiekan valumiseen reiän kautta. Tulikello toimii palavan nesteen avulla, jolloin tulen palaessa nestettä kuluu ja nestesäiliön pinnantasoinnansäilyy.



kellonajan. Kynttiläkello toimii tulikellon tavoin, jolloin kynttilän pituudesta voidaan määrittää aika. [2, s. 74.]



*Kuva 1. Yksinkertainen aurinkokello [4.]*

Aluksi mekaanisten kellojen kellotaulu oli jaettu 24:ään osaan ja aikaa osoitti vain yksi osoitin [2, s. 75]. Myöhemmin kelloihin yleistyi 12-osainen kellotaulu ja ajan ilmaiseminen useamman viisarin avulla. Mekaanisten kellojen toiminta perustuu yleensä maan vetovoimaan. Mekaanisten kellojen jälkeen tulivat kellot, jotka toimivat sähköllä. Sähköisten kellojen ytimenä toimii yleensä kvartsikide. Nykyään on käytössä erittäin tarkkoja atomikelloja, jotka mittaavat aikaa atomin värähtelytaajuuden avulla.

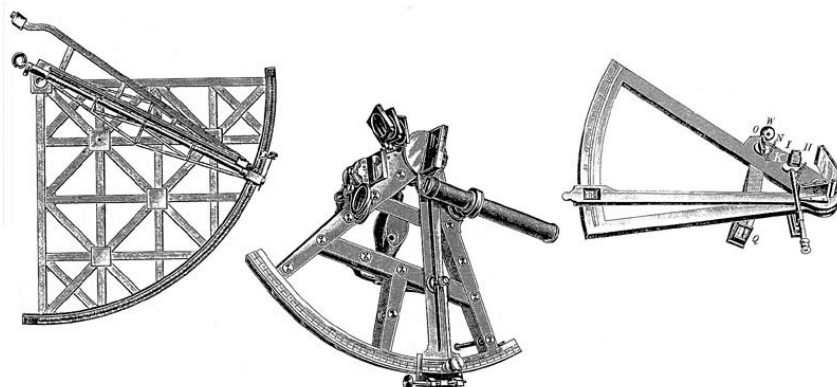
## 2.4 PC:n kello

PC-tietokoneissa on kello, joka tunnetaan nimellä RTC (Real Time Clock). Siihen voidaan tallentaa kellonaika ja päivämäärä. Piiri on paristovarmennettu, jolloin tiedot säilyvät, vaikka tietokone sammutetaan. Kun tietokone käynnistetään, käyttöjärjestelmä hakee tietokoneen RTC-ajan ja alkaa päivittää aikaa keskeytysten avulla. Kun järjestelmää kuormitetaan, käyttöjärjestelmä ei enää ehdi päivittää kelloaan, jolloin käyttöjärjestelmän ylläpitämä kellonaika alkaa jätättää. Tämän takia on kehitetty menetelmiä, joilla kellonaika saadaan pysymään oikeana.

Yksi näistä ratkaisuista on NTP (Network Time Protocol). NTP-palvelimet ylläpitävät kellonaikaa ja jakavat sitä muille tietokoneille. Työasemassa on ohjelma, joka kysyy kellonajan tällaiselta palvelimelta. Yleensä kellonaika tarkastetaan useammalta palvelimelta, joiden ajoista lasketaan keskiarvo, jolloin päästään melko suuriin tarkkuuksiin.

## 2.5 Navigointi

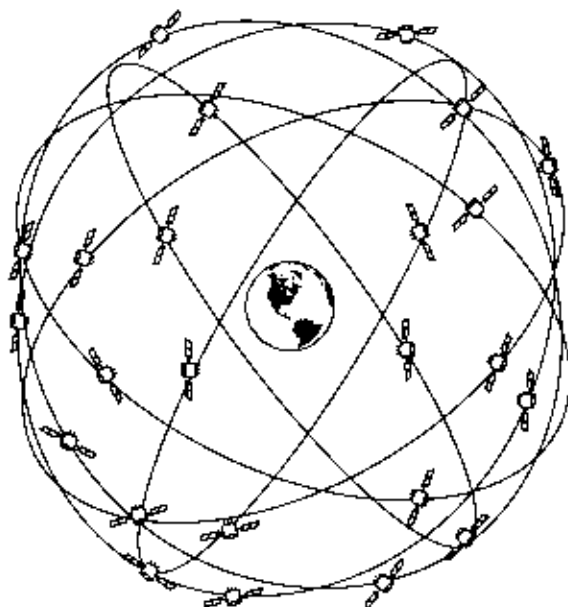
Navigoinnilla tarkoitetaan sijainnin, suunnan ja etäisyyden määrittämistä sekä liikkumista kohteeseen. Paikan määrittäminen on tärkeä asia merenkulussa ja ilmailussa. Ennen satelliittipaikannusta merenkulussa on suunnistettu muun muassa maamerkkien, kartan, kompassin, sekstantin ja tähtikuvioiden mukaan. Sekstantilla voidaan määrittää leveysaste mittaamalla Auringon tai jonkin muun planeetan korkeus taivaanrannasta. Myös pituusaste voidaan määrittää taivaan kappaleiden mukaan, jos tiedossa on tarkka kellonaika. Kuvassa 2 on esitetty erilaisia kulmanmittaukseen käytettyjä laitteita.



*Kuva 2. Erilaisia kulmanmittaukseen käytettyjä laitteita [5.]*

Nykyään käytössä on satelliittipaikannusjärjestelmä GPS (Global Positioning System). GPS-järjestelmä koostuu kolmesta eri osasta, jotka ovat satelliitit, tarkkailuasemat ja GPS-paikantimet. Järjestelmän toiminta perustuu etäisyyden mittaukseen. Järjestelmään kuuluu 24 satelliittia, ja ne kiertävät maapalloa kuudella eri kiertoradalla, joiden väli on 60 astetta [6, s. 27]. Satelliitit kiertävät maapalloa keskimäärin 20183 kilometrin etäisyydellä maapallon pinnasta ja nii-

den kiertoaika on 11 tuntia ja 58 minuuttia [6, s. 28]. Joka hetki maapallon pinnalta katsottuna satelliitteja on näkyvissä vähintään kuusi [7, s. 11]. Kuvassa 3 on esitetty, kuinka satelliitit kiertävät maapalloa.



*Kuva 3. GPS-satelliitit kiertoradalla [8.]*

Satelliitit lähettävät signaaleja, jolloin maan päällä oleva GPS-paikannin voi laskea signaalin kulkuajasta etäisyyden satelliittiin. Satelliitit tunnistetaan toisistaan näennäissatunnaisen signaalin avulla (PRN, Pseudo Random Noise Code) [7, s. 119]. Signaalin kulkuajan mittaamiseksi on satelliitin ja GPS-paikantimen kellojen oltava täsmälleen samassa ajassa. Satelliittien kelloja verrataan jatkuvasti maassa olevien tarkkailuasemien atomikelloihin. Kolmen satelliitin avulla on mahdollista laskea paikkatieto, mutta jos halutaan tietää myös korkeus, satelliitteja tarvitaan neljä. Nopeuden mittaaminen GPS:llä tapahtuu käyttäen hyväksi Doppler-ilmiötä. Syvällisempää tietoa GPS-järjestelmän toiminnasta löytyy Markku Poutasen kirjasta GPS-paikanmääritys [7].

Nykyäänkin merenkulun ammattilaiset osaavat navigoida ilman satelliittipaikannusjärjestelmää. Jos GPS-paikannin rikkoontuu oltaessa avomerellä, on sijainti silti pystyttävä määrittämään tarkasti.

### 3 KYTKENTÄ

Kellonajan jakolaite on rakennettu Kajaanin ammattikorkeakoulun opetuskäytössä olevalle prosessorikortille (AMKTEL1 1.12.1997). Piirilevyllä on valmiina johdotukset prosessorille, osoitedekoodaukselle ja muistipiirille.

Laitteeseen tarvittavat komponentit ja osat löytyivät ammattikorkeakoulun laborioiden komponenttivalikoimista lukuun ottamatta sarjaliikennemuunninta. Sen tilaamisen on hoitanut kehitysinsinööri Markku Karppinen. Kytkennässä on käytetty liitteen A mukaisia osia ja komponentteja.

Kellonajan jakolaitteen kytkentä on esitetty liitteessä B. Kytkennät kortille on tehty räppäämällä. Räppäämisellä tarkoitetaan komponenttien yhdistämistä langalla, joka kierretään kortilla oleviin tappeihin siihen suunnitellun työkalun avulla.

#### 3.1 Laitteelle asetetut vaatimukset

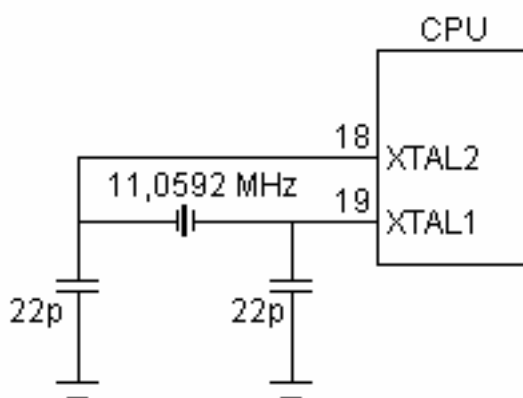
Laitteen pitää lukea satelliittisynkronoidusta kellosta sekunnin pulssia ja päivittää omaa kelloaan. Kellonaika ja päivämäärä on pystyttävä lähettämään ammattikorkeakoulun tietoverkkoon minuutin välein. Lisäksi työasemaan on tehtävä ohjelma, joka lukee ajan ja päivämäärän.

Laitetta pitää pystyä ohjaamaan PC-tietokoneen terminaaliohjelmalla sarjaportin kautta. Laitteeseen on pystyttävä asettamaan kellonaika ja päivämäärä sekä myös lukemaan ne tarvittaessa. Kellolle on pystyttävä lisäämään tarvittaessa karkaussekunti joului- tai kesäkuun viimeisenä päivänä.

### 3.2 Prosessori

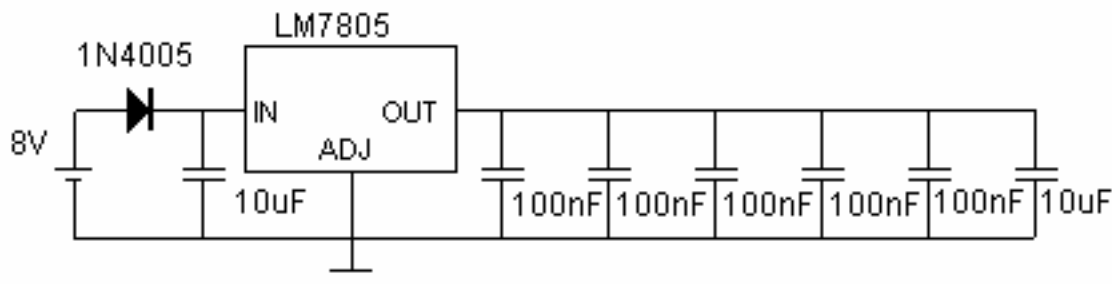
Prosessorina laitteessa käytettiin Intelin valmistamaa mikroprosessoria 8051. 8051-prosessorin pääominaisuuksiin kuuluu 8-bittinen CPU, 4x8 kaksisuuntaista I/O-porttia sekä kaksisuuntainen sarjaportti. Joillakin prosessorin kytkentänoimilla on useampia toimintoja. Kaikkia toimintoja ei ole mahdollista käyttää yhtä aikaa.

Prosessori tarvitsee toimiakseen ulkoisen kiteen, koska sitä ei ole integroitu prosessoriin. Kytkennässä käytettiin 11,0592 MHz:n kidettä, jonka avulla sarjaportin nopeudeksi saatiin melko tarkasti nopeus 9600 bit/s. Kuvassa 4 on esitetty, kuinka kide on kytketty prosessoriin.



*Kuva 4. Prosessorin kiteen kytkentä*

8051-prosessori tarvitsee toimiakseen 5 V:n käyttöjännitteen. Kuvan 5 kytkentä alentaa käytettävän käyttöjännitteen 5 V:iin jänniteregulaattorin LM7805 avulla. Myös muut komponentit kellonajan jakolaitteen kytkennässä käyttävät 5 V:n jännitettä. Kytkennässä olevat kondensaattorit ovat by-pass-kondensaattoreita, jotka poistavat käyttöjännitteen häiriöitä. Kytkennässä käytetty diodi suojaa laitetta, kun käyttöjännitteen napaisuus on väärinpäin kytketty.



Kuva 5. Jännitteen alentaminen

Tässä työssä käyttöjännitettä on syötetty laitteeseen ammattikorkeakoulun laborioiden työpöydissä olevilla jännitelähteillä. Niiden jännitenäytön tarkkuus ei ole paras mahdollinen, joten kuvan 5 mukaisen kytkennän käyttäminen on perusteltua. Tämän kytkennän ansiosta laitteen jännitelähteenä voidaan käyttää myös esimerkiksi 9 V:n paristoa.

### 3.3 Muistipiiri

Ohjelmamuistina käytetään M27C256B EPROM -muistia (Electrically Erasable Programmable Read Only Memory), jonka kapasiteetti on 256 kb. Piiri voidaan ohjelmoida useita kertoja, mutta ohjelmointikertojen välillä piiri on tyhjennettävä ultraviolettivalolla.

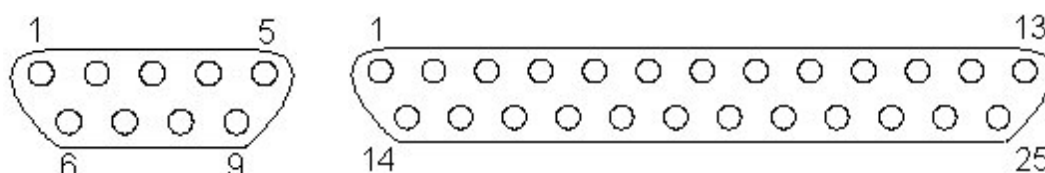
Kytkenässä on käytetty latch-piiriä 74HC373, koska porttia P0 käytetään saman lukujakson aikana osoitteen ja tiedon lähettämiseen. Prosessorin ALE-linjalla (Address Latch Enable) määrätään hetki, jolloin osoite lukitaan latch-piiriin. Tämä kytkentä on esitetty liitteessä B.

Kun prosessorin  $\overline{EA}$ -tulo (External Access) on kytkettynä maahan, niin ohjelmamuisti on kokonaan ulkoista ROM:ia. Muistipiirin  $\overline{CE}$ -linja (Chip Enable) ja  $\overline{OE}$ -linja (Output Enable) on oltava aktiivisia lukuaikana.  $\overline{CE}$ -linja on kytketty maahan.  $\overline{OE}$ -linja on kytketty prosessorin  $\overline{PSEN}$ -linjaan (Program Store Enable), jota käytetään ulkoisen muistin lukupulssina. [9, s. 52.]

### 3.4 Sarjaliikenne

RS-232C määrittelee yleisesti käytössä olevan sarjaliikennöimistavan. Se sisältää neljä osaa, jotka ovat sähköiset signaaliominaisuudet, liitännän mekaaniset ominaisuudet, liitäntäpiirien toimintakuvaus ja standardiliitännät. [10.]

Sarjaliikenteessä yleisesti käytettyjä nopeuksia ovat 300, 1200, 2400, 4800, 9600, 19200, 38400 ja 115200 bit/s. Liittiminä käytetään yleensä 9- tai 25-napaista liitintä. PC-tietokoneissa on yleensä 9-napainen urosliitin. Kuvassa 6 on esitetty 9- ja 25-napaisten liittimien nastanumeroinnit.



Kuva 6. Liittimien nastanumeroinnit takaapäin katsottuna

Taulukossa 1 on esitetty 9-napaisen liittimen nastojen käyttötarkoitus sekä niistä käytettävät lyhenteet. Taulukossa on ilmoitettu myös englanninkielinen nimitys, jotta lyhenteet olisi helpompi ymmärtää.

Taulukko 1. D-9-liittimen nastaselitteet [10 ja 11]

Nasta	Lyhenne	Englanninkielinen nimitys	Suomenkielinen nimitys
1	DCD	Data Carrier Detect	Kantoaallon tunnistus
2	RXD	Receive Data	Datan vastaanotto
3	TXD	Transmit Data	Datan lähetys
4	DTR	Data Terminal Ready	Päätelaitteen valmius
5	SG	Signal Ground	Signaalimaa
6	DSR	Data Set Ready	Siirtolaitteen valmius
7	RTS	Request To Send	Lähetyspyyntö
8	CTS	Clear To Send	Lähetysvalmius
9	RI	Ring Indicator	Soiton osoitus

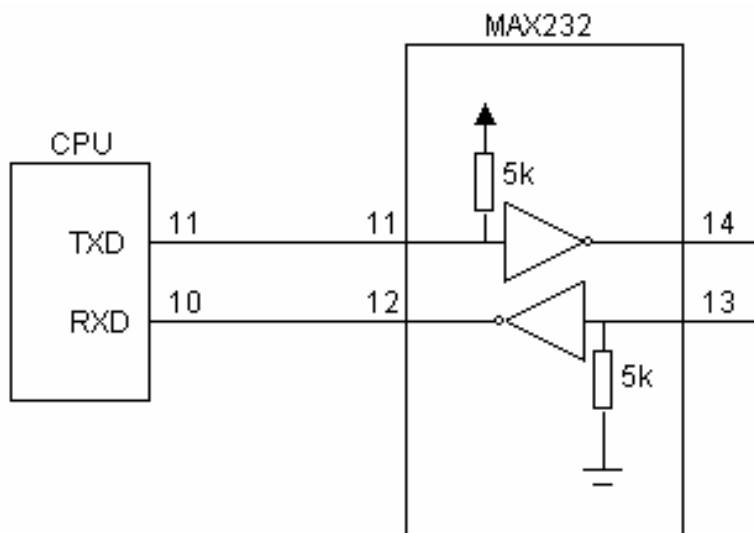
D-25-liittimen nastajärjestys poikkeaa D-9-liittimen järjestyksestä, kuten taulukosta 2 huomataan (vertaa taulukkoon 1).

*Taulukko 2. D-25-liittimen nastaselitteet [11 ja 12]*

Nasta	Lyhenne	Englanninkielinen nimitys	Suomenkielinen nimitys
1	GND	Chassis Ground	Suojamaa
2	TXD	Transmit Data	Datan lähetys
3	RXD	Receive Data	Datan vastaanotto
4	RTS	Request To Send	Lähetyspyyntö
5	CTS	Clear To Send	Lähetysvalmius
6	DSR	Data Set Ready	Siirtolaitteen valmius
7	SG	Signal Ground	Signaalimaa
8	DCD	Data Carrier Detect	Kantoaallon tunnistus
9		Test Voltage +	Testijännite +
10		Test Voltage -	Testijännite -
11		Unassigned	Ei määritetty
12	SDCD	Secondary DCD	Toissijainen DCD
13	SCTS	Secondary CTS	Toissijainen CTS
14	STXD	Secondary TXD	Toissijainen TXD
15	TC	Sync TX Timing DCE	Lähetyskello DCE
16	SRXD	Secondary RXD	Toissijainen RXD
17	RC	Sync RX Timing DCE	Vastaanottokello DCE
18		Unassigned	Ei määritetty
19	SRTS	Secondary RTS	Toissijainen RTS
20	DTR	Data Terminal Ready	Päätelaitteen valmius
21	SQ	Signal Quality Detect	Signaalin laadun valvonta
22	RI	Ring Indicate	Soiton osoitus
23	DRS	Data Signal Rate	Nopeuden valinta
24	XTC	Sync TX Timing DTE	Lähetyskello DTE
25		Unassigned	Ei määritetty



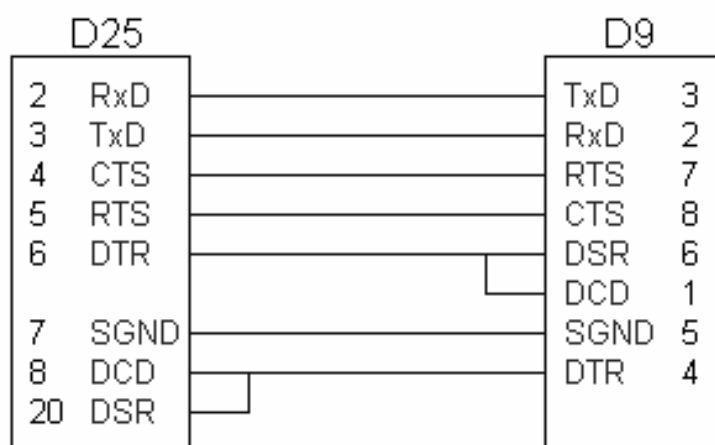
RS-232-signaalien sovittamiseen prosessorin TTL-tasoiseen tuloon/lähtöön käytettiin Maxim 232 -piiriä. MAX232-piiri on yleisesti käytetty komponentti signaalitasojen sovittamiseen. Kuvassa 7 on esitetty kytkentä, jolla signaalien sovittaminen tehtiin.



*Kuva 7. Signaalien sovittaminen*

#### 4 SARJALIIKENNEMUUNNIN

Kytkenässä käytettiin Moxa DE-211 -sarjaliikennemuunninta. Se muuttaa sarjamuotoisen tiedon pakettimuotoon ja lähettää sen tietoverkkoon. Muuntimeen voidaan asettaa IP-osoite sekä muut tarvittavat tiedot terminaaliohjelmalla. Laitteen asetusten muuttamiseksi tarvittiin kuvan 8 mukainen sarjakaapeli.



Kuva 8. Sarjakaapeli laitteen asetusten muuttamiseksi [13.]

Terminaaliohjelman voi käyttää esimerkiksi Microsoft Windowsin mukana tulevaa HyperTerminal-ohjelmaa. Terminaaliohjelman asetukset ovat 19200, N, 8, 1. Vuonohjaus ei saa olla kytkettynä. Laitteeseen otettiin yhteys terminaaliohjelmalla sarjaliikennemuuntimen valintakytkimen 1 ollessa ON-asennossa. Kytkimet 2, 3 ja 4 olivat asennossa OFF, koska käytetään RS-232:sta. Kytkinten merkityksen voi todeta taulukosta 3.

Taulukko 3. Laitteen valintakytkimien merkitys [13.]

SW1	Serial Connection	SW2	SW3	SW4	Serial Interface Mode
		OFF	OFF	OFF	RS-232
ON	RS-232 Console	OFF	ON	ON	RS-422
		ON	OFF	ON	4-Wire RS-485 by RTS
OFF	Data Comm	ON	ON	ON	4-Wire RS-485 by ADDC
		ON	OFF	OFF	2-Wire RS-485 by RTS
		ON	ON	OFF	2-Wire RS-485 by ADDC

Yhteyden muodostamisen jälkeen asetukset muutettiin taulukon 4 mukaisiksi. Tässä tapauksessa asetukset on asetettu siten, että tietoa lähetetään broadcast-osoitteeseen 10.0.0.255. Laitteelle on valittu IP-osoitteeksi privaattiosoite, koska nämä osoitteet eivät reitity eteenpäin. Kun laitteella halutaan lähettää tietoa tietoverkkoon, täytyy valintakytkimen 1 olla asennossa OFF taulukon 3 mukaisesti.

*Taulukko 4. Moxa DE-211 asetukset*

Server Config	
Server Name	TEST
IP Address	10.0.0.5
Netmask	255.255.255.0
OP_Mode	
Application	UDP Server/Client
Destination IP	10.0.0.255
Destination Port	4000
Serial Port	
Baud Rate (bps)	9600
Parity	None
Data Bits	8
Stop Bit	1
Flow Control	None
UART FIFO	Enable

## 5 OHJELMA PROSESSORILLE

Ohjelma toteutettiin C-kielellä käyttäen apuna IAR-kehitysympäristöä. Ohjelmaa kirjoitettaessa testaaminen tapahtui Emul-51-emulaattorin avulla. Emulaattoria käytettäessä prosessorikortilla ei tarvita muistipiiriä eikä prosessoria. C-kielinen ohjelmakoodi on esitetty liitteessä D.

### 5.1 Sarjaportti

Sarjaportilla on useita eri toimintamuotoja ja se valitaan SCON-rekisterin SM0- ja SM1-biteillä. Tässä ohjelmassa käytettiin toimintamuoto 1:stä, jolloin SM0 = 0 ja SM1 = 1. Vastaanotto on sallittu, kun REN = 1 ja vastaanotetaan START-bitti. Lähetys- ja vastaanottorekisterit näkyvät SBUF-rekisterinä vaikka ovatkin todellisuudessa kaksi eri rekisteriä. Lähetys käynnistyy, kun SBUF-rekisteriin kirjoitetaan jotakin. [9, s. 109.]

Sarjaportti toimii keskeytysten avulla. Keskeytyksiä voidaan helposti sallia ja estää IE-rekisterin avulla. IP-rekisterillä määritellään keskeytysten prioriteetti. TCON-rekisterillä määritellään keskeytyksien tyypit ja PCON-rekisterillä ohjataan tehonsyöttöä. TMOD-rekisterillä valitaan laskureiden/ajastimien toimintamuoto. Rantalan dokumentissa Intel MCS-51 -perhe [14] on selvitetty rekisterien arvojen muodostumiset.

Toimintamuoto 1:ssä on käytössä 10 bitin kehys (START+8 bit+STOP). Lähetys- ja vastaanottonopeus on muunneltavissa TH1-rekisterin avulla. TH1:n arvo lasketaan kaavan 1 avulla.

$$TH1 = 256 - \frac{\frac{kellotaajuus}{384}}{baudinopeus} \quad (1)$$

Tässä ohjelmassa nopeutena käytettiin 9600 bit/s. Kytkenässä käytettiin 11,0592 MHz:n kidettä, koska sen avulla saadaan melko tarkasti muodostettua nopeus 9600 bit/s. Näillä arvoilla TH1:n arvoksi saadaan 253, joka on heksalukuna esitettynä FD.

## 5.2 Toiminta

Ensimmäisenä ohjelmassa ladataan tarvittavat header-tiedostot sekä alustetaan muuttujat, laskuri ja sarjaportti. Ohjelma perustuu ikuiseen silmukkaan, jossa ohjelman suoritus viipyy, ja odottaa keskeytystä. Ohjelma odottaa ulkoiseen keskeytykseen  $\overline{INT0}$  saapuvia pulsseja, joiden avulla se päivittää kelloaan ja kalenteriaan. Laitteessa on kaksiasentoinen kytkin, jolla voidaan valita laitteen tila. Ohjelma lähettää kellonajan ja päivämäärän sarjaportin kautta sarjaliikennemuuntimelle minuutin välein, jos kytkin on asennossa 0. Kun kytkin on asennossa 1, laitteeseen voidaan muuttaa asetuksia terminaaliohjelmalla sarjaportin kautta. Ohjelman kalenteri toimii oikein vuoteen 2100 asti. Ohjelman toiminnallinen vuokaavio on esitetty liitteessä C.

## 5.3 Muistipiiri

Ohjelman valmistuttua se tallennettiin EPROM-muistipiirille ammattikorkeakoulun laboratoriossa olevan ohjelmointilaitteen avulla. Muistipiiri liitettiin prosessorikortilla valmiina olevaan kantaan. Prosessorikortille kiinnitettiin myös prosessori, jonka jälkeen laite toimi itsenäisesti ja emulaattorilaitteistoa ei enää tarvittu.

## 6 OHJELMA TYÖASEMALLE

Ohjelma muuttaa PC-tietokoneen kellonajan ja päivämäärän. Se on tehty C++-ohjelmointikielellä käyttäen Borland C++ 5.02 -kääntäjää. Ohjelmassa käytettiin Microsoftin WinSock-rajapintaa. WinSock on ohjelmointirajapinta, joka perustuu BSD-socketteihin (Berkeley Software Distribution). Ohjelma kuuntelee porttia 4000 (UDP). C++-kielinen ohjelma on esitetty liitteessä F.

### 6.1 Protokollat

Protokolla on ennalta sovittu tapa, josta selviää kuinka jokin asia tapahtuu. Protokollia on kahdenlaisia, kuljetus- ja siirtoprotokollia. Seuraavassa on esitelty kuljetusprotokollia.

TCP (Transmission Control Protocol) on yhteydellinen protokolla. Lähetettävät paketit on numeroitu. Kun vastaanottajalle saapuu paketti, niin se kuitataan vastaanotetuksi, jolloin lähettäjä varmistuu tiedon perillemenosta.

UDP (User Datagram Protocol) on yhteydetön protokolla. Paketteja ei ole numeroitu kuten TCP:ssä, jolloin vastaanottaja ei myöskään voi kuitata paketteja vastaanotetuiksi. Tämän takia UDP on melko epävarma kuljetusprotokolla.

ICMP (Internet Control Message Protocol) on hieman harvinaisempi protokolla, josta esimerkkinä mainittakoon komento PING. PING-komento lähettää echo request -sanoman, jonka jälkeen vastaanottaja lähettää vastaukseksi echo replay -sanoman. [15.]

Tässä työssä protokollana on käytettiin UDP:tä, koska kellonaikaa ja päivämäärää lähetetään useammalle tietokoneelle. TCP:ssä voidaan lähettää vain unicastia, mutta UDP:ssä on mahdollisuus multicast- ja broadcast-lähetyksiin.

## 6.2 Socket

Socketilla tarkoitetaan IP-osoitteen ja porttinumeron (TCP tai UDP) muodostamaa paria. Esimerkiksi IP-osoite 127.0.0.1 ja FTP-protokollan käyttämä portti 21 muodostavat socketin. Tätä tietoa voidaan käyttää sovelluksen tunnistamiseen. [15.]

Porttinumero voi olla 0–65535. Alle 1024 olevat portit ovat varattuja portteja. Yli 1024 olevat portit ovat vapaammin käytettävissä, mutta näissäkin on varattuja portteja. Taulukossa 5 on esitetty muutamia yleisiä palveluita sekä niiden käyttämiä portteja.

*Taulukko 5. Varattuja portteja*

Palvelun lyhenne	Palvelu	Portti
FTP	File Transfer Protocol	21
SSH	Secure SHell	22
TELNET	TELEcommunications NETwork protocol	23
SMTP	Simple Mail Transfer Protocol	25
DNS	Domain Name System	53
WWW	World Wide Web	80
POP3	Post Office Protocol version 3	110
NTP	Network Time Protocol	123

## 6.3 Toiminta

Ensimmäisenä ohjelmassa ladataan tarvittavat kirjastot, määritellään WinSocketin tyyppi sekä käytettävä protokolla, joka on tässä ohjelmassa UDP. Sen jälkeen luodaan socket ja odotetaan tietoa porttiin 4000. Datan saavuttua tutkitaan, onko vastaanotettu tieto kellonaika vai päivämäärä. Kellonaika ja päivämäärä muutetaan järjestelmään Windowsin käskyjen "time" ja "date" avulla. Tämän jälkeen ohjelman suoritus alkaa alusta. Ohjelma pyörii ikuisessa silmukassa. Ohjelman toiminnallinen vuokaavio on esitetty liitteessä E.

## 7 TESTAAMINEN

Laitteen toiminta on testattu Kajaanin ammattikorkeakoulun laboratoriossa. Laitteelle syötettiin sekunnin pulssia GPS-laitteelta. Kellonajan jakolaite käynnistettiin ja siihen asetettiin kellonaika ja päivämäärä. Laitteen toimintatarkkuuden testaamista varten se jätettiin yön ajaksi käyntiin. Noin 24 tunnin aikana laitteen käynti oli alle yhden sekunnin. Täten voidaan todeta laitteen kellon tarkkuuden olevan hyvä.

Karkauspäivä testattiin laittamalla päivämääräksi 28.2.2004 ja kellonajaksi 23:59:59. Todettiin päivämäärän muuttuvan oikein, eli seuraava päivä oli 29.2.2004. Testi suoritettiin myös päivämäärällä 28.2.2005, jolloin seuraava päivä oli 1.3.2005. Myös karkaussekunnin toiminta tarkastettiin samantapaisella metodilla ja sen todettiin toimivan.

Laite kytkettiin verkkoon ja kahdella PC-tietokoneella oli käynnissä C++-kielinen ohjelma, joka kuunteli porttia 4000 (UDP). Ohjelma vastaanotti kellonajan ja päivämäärän sekä päivitti tietokoneen kelloa ja kalenteria näiden tietojen pohjalta. Laitteen käyttöohje on esitetty liitteessä G.



## 8 JATKOKEHITYS

Tätä insinööriötä voidaan pitää kellonajan jakolaitteen prototyyppinä, jolle on monia jatkokehitysmahdollisuuksia.

Nykyisen prosessorikortin voisi korvata kokonaan syövytetyllä piirilevyllä sekä laitteen voisi koteloida. Laitteella olisi hyvä olla myös akkuvarmennus mahdollisten sähkökatkojen varalle sekä akkujen automaattinen lataus normaalitilassa.

Prossessorin C-kielistä ohjelmaa voisi jatkokehittää ja paneutua varsinkin kellon toimintatarkkuuden parantamiseen. PC-tietokoneen ohjelmaa voisi kehittää käyttökelpoisemmaksi. Tämänhetkinen C++-kielinen ohjelma ei sovellu sellaiseenaan ajettavaksi ammattikorkeakoulun työasemissa. Ohjelmassa on puutteelliset tarkastusalgoritmit vastaanotettavan tiedon ja lähettäjän suhteen eikä ohjelma sulaudu Windows-ympäristöön.

Laitteen voisi toteuttaa myös ilman prosessoriohjausta, jolloin laskuri laskisi GPS:ltä tulevia sekunnin pulsseja. Pulssin 60 tullessa kytkentä ohjaisi GPS:n sarjaliikenteen sarjaliikennemuuntimelle, joka lähettäisi GPS:n tulosteen Kajaa-nin ammattikorkeakoulun tietoverkkoon. Tämän jälkeen laskuri nollattaisiin ja jälleen minuutin kuluttua tietoa lähetettäisiin tietoverkkoon. Tällä menetelmällä saavutettaisiin se etu, että laitteeseen ei tarvitsisi asettaa mitään asetuksia manuaalisesti, jolloin laitteen toimintatarkkuus paranisi. Tämän johdosta työasemassa olevaan ohjelmaan täytyisi tehdä funktio, joka kerää GPS:n tulosteesta oleellisen, kellonajan ja päivämäärän.

## 9 YHTEENVETO

Työssä suunniteltiin ja rakennettiin laite, joka lähettää kellonajan sekä päiväyksen Kajaanin ammattikorkeakoulun tietoverkkoon. Laite rakennettiin opetus- käytössä olevalle prosessorikortille, jossa käytettiin Intel 8051 -prosessoria. Kellonajan ja päivämäärän lähettäminen tietoverkkoon tapahtui Moxa-merkkisen sarjaliikennemuuntimen avulla, joka muuntaa sarjamuotoisen datan pakettimuotoiseksi ja lähettää sen tietoverkkoon.

Työkokonaisuuteen kuului myös tehdä työasemaan ohjelma, joka lukee kellonajan ja päivämäärän. Ohjelma toteutettiin C++-ohjelmointikielellä käyttäen apuna WinSock-rajapintaa, jonka avulla on yksinkertaista toteuttaa verkkoa käyttävä ohjelma.

Kellonajan jakolaitteen toiminta testattiin Kajaanin ammattikorkeakoulun laboratoriossa, jolloin laitteen käynniksi mitattiin noin yksi sekunti vuorokautta kohti. Laitteen kalenteri ja kello toimivat erinomaisesti, ottaen huomioon karkausvuoden, -päivän ja -sekunnin. Testitulokset olivat odotustenmukaisia.

Työ oli kokonaisuudessaan mielenkiintoinen ja monipuolinen. Erityisen kiinnostavaa oli PC-tietokoneelle tulevan ohjelman suunnittelu ja toteuttaminen, koska minulla ei ollut aiempaa kokemusta verkkoa käyttävästä ohjelmasta.

Tämä työ on luettavissa Kajaanin ammattikorkeakoulun kirjastossa, josta sen voi myös lainata. Työ on saatavilla sähköisessä muodossa Kajaanin ammattikorkeakoulun kirjaston Kaktus-tietokannasta.

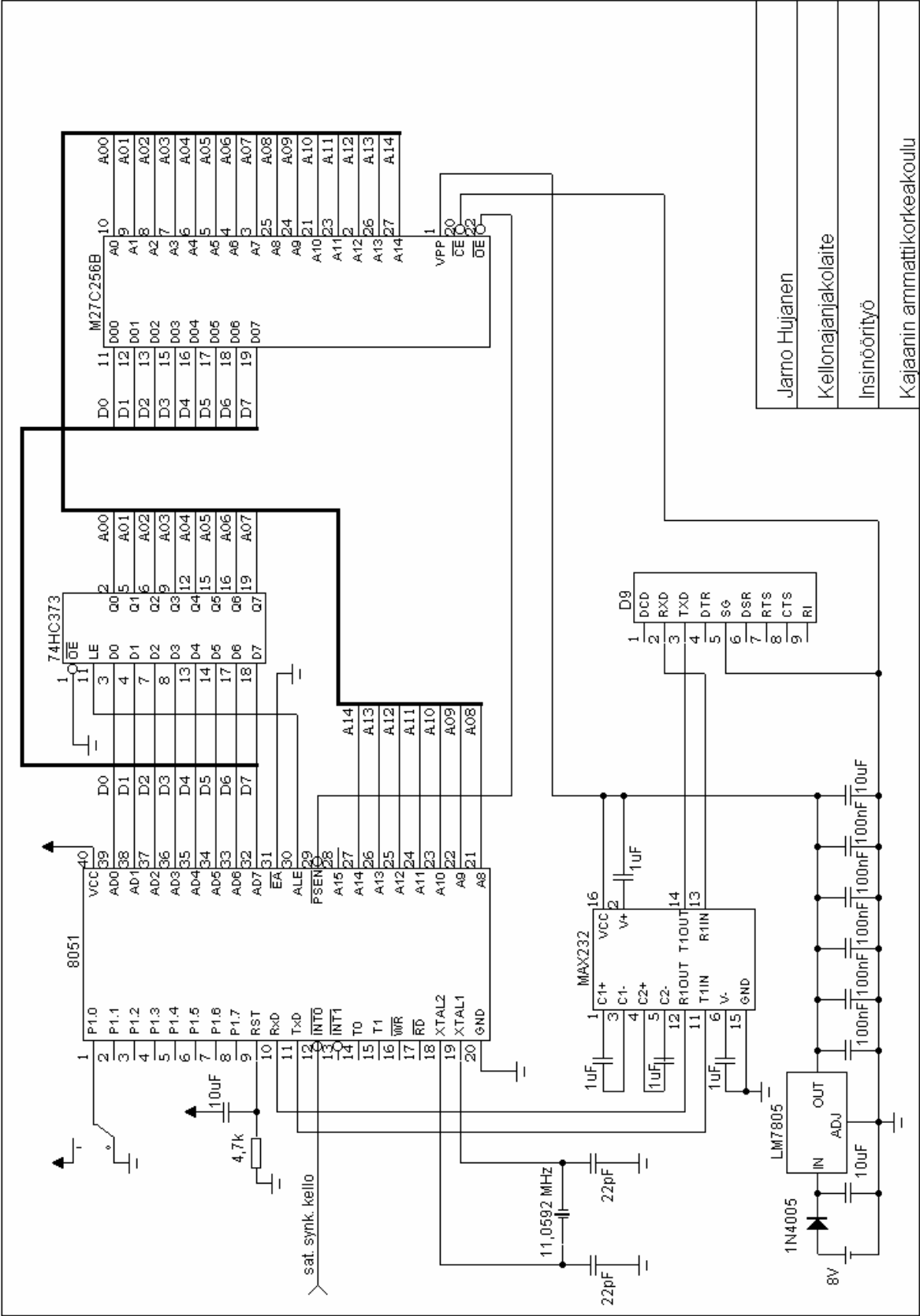
## LÄHDELUETTELO

- 1 Eskola, M. Weilin+Göösin tietosanakirja. Gütersloh: Mohndruck Graphische Betriebe GmbH, 1994. ISBN 951-35-5066-4.
- 2 Oja, H. Aikakirja. Keuruu: Otavan Kirjapaino Oy, 1999. 288 s. ISBN 951-1-16334-5.
- 3 Sekunti. Kaleva. 11.12.2003.
- 4 Lastenmaa. Luettu 16.3.2004. [WWW-dokumentti].  
<<http://www.credo.fi/lastenmaa/tiede/kelloja.htm>>
- 5 Rekola, R. Tähtitieteen peruskurssi I. 2002. Luettu 26.3.2004.  
[WWW-dokumentti].  
<<http://www.astro.utu.fi/kurssit/ttpk1/ttpkl/16Kaukoputki.html>>
- 6 Miettinen, S. GPS käsikirja. Karisto: Genimap Oy, 2002. 166 s. ISBN 951-593-786-8.
- 7 Poutanen, M. GPS-paikanmäärittäminen. Hämeenlinna: Karisto Oy, 1998. 269 s. ISBN 951-9269-89-4
- 8 Dixon, J. Mobile Robot Navigation. 20.5.1997. [WWW-dokumentti].  
<[http://www.doc.ic.ac.uk/~nd/surprise\\_97/journal/vol1/jmd/](http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol1/jmd/)>
- 9 Rantala, P. Mikrotietokonetekniikka, tietokonetekniikka osa 2. Oulu: Tietokotka, 2001. 117 s. ISBN 951-559-258-5.
- 10 Campbell, J. RS-232 liitäntä. Mänttä: Mäntän Kirjapaino, 1985. 178 s. ISBN 951-35-3384-0

- 11 RS232 Communication Configuration. Luettu 23.2.2004. [WWW-dokumentti].  
<[http://members.tripod.com/~oldboard/assembly/rs232\\_pins.html](http://members.tripod.com/~oldboard/assembly/rs232_pins.html)>
- 12 RS-232C sarjaliitäntä IBM PC/AT:ssa ja yhteensopivissa. Muutettu 20.7.1993. [WWW-dokumentti]. <<http://www.hut.fi/~then/mytexts/rs-232c.html>>
- 13 NPort Express Hardware Installation Guide for DE-211. Muutettu 28.11.2002. [PDF-dokumentti].  
<[http://www.moxa.com/doc/manual/nport/DE211/HW3/npde211\\_v3.pdf](http://www.moxa.com/doc/manual/nport/DE211/HW3/npde211_v3.pdf)>
- 14 Rantala, P. Intel MCS-51 -perhe. Muutettu 24.9.2001. [PDF-dokumentti]. <<http://www.tietokotka.planet.fi/tiedostot/mcs51.pdf>>
- 15 Kaario, K. TCP/IP-verkot. Porvoo: WS Bookwell, 2002. 396 s. ISBN 951-846-107-4

## Komponenttiluettelo

Määrä	Komponentti	Arvo / tyyppi
2	Kondensaattori	22 pF
4	Kondensaattori	1 µF
5	Kondensaattori	100 nF
3	Kondensaattori	10 µF
1	Vastus	4,7 kΩ
1	Proessori	Intel 8051
1	Kide	11,0592 MHz
1	Latchpiiri	74HC737
1	EPROM muistipiiri 256 kb	M27C256B
1	Sarjaliikenneohjain	Maxim 232
1	Jänniteregulaattori	LM7805
1	Diodi	1N4005
1	Kytkin	2-asentoinen
1	Uros-liitin	D9
1	Proessorikortti	AMKTEL1 1.12.1997
1	Sarjaliikennemuunnin	Moxa DE-211

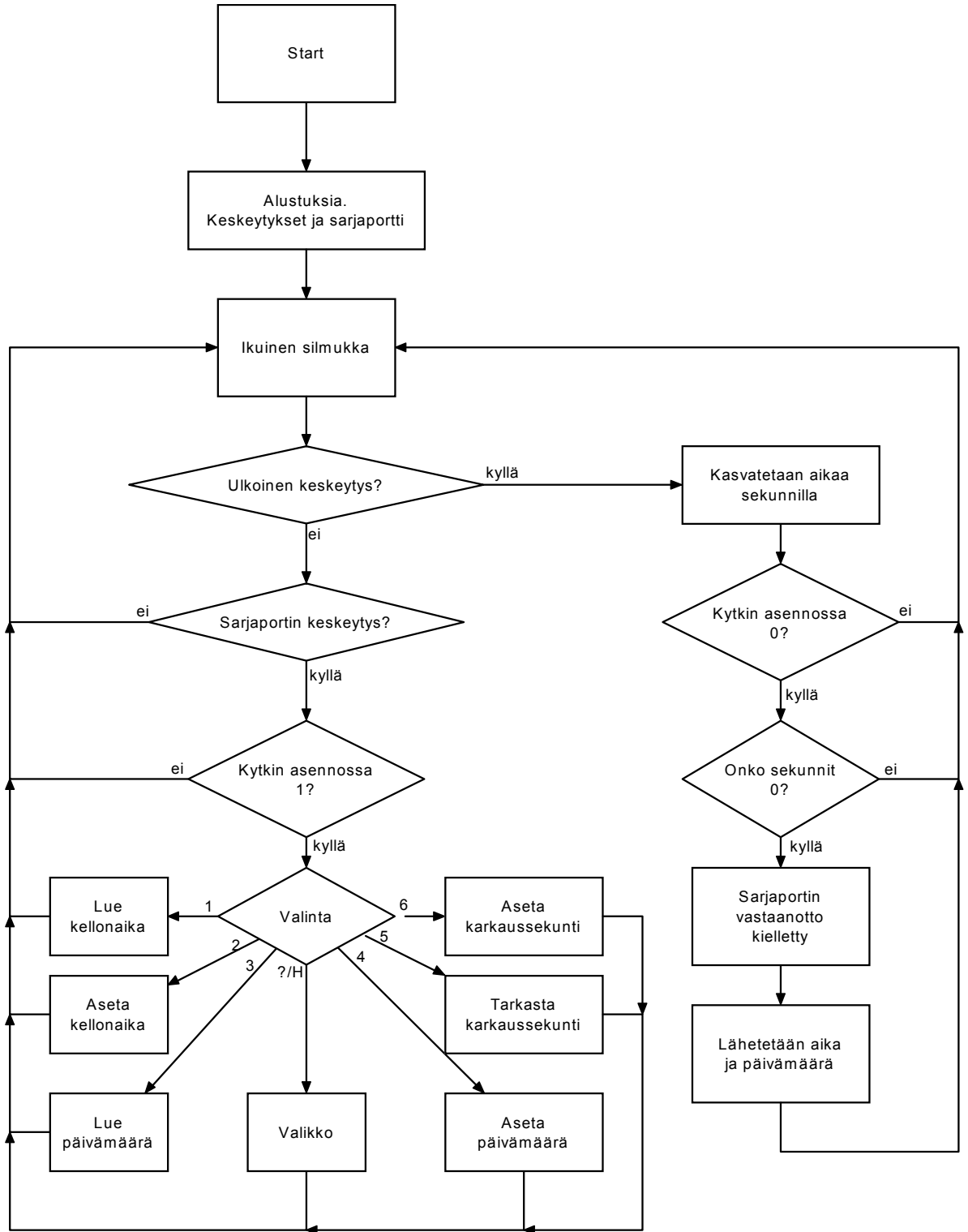


Jarmo Hujanen

Kellonajanjakolaite

Insinööri

Kajaanin ammattikorkeakoulu



#Jarno Hujanen  
#Insinöörityö  
#Kellonajanjakolaite  
#Kajaanin ammattikorkeakoulu

#Ohjelma laskee 8051:n ulkoiseen keskeytykseen (INT0)  
#sekunnin välein tulevia pulsseja ja päivittää kelloaan.  
#Kellonaika ja päivämäärä lähetetään 8051:n sarjaportin  
#kautta sarjaliikennemuuntimelle.

```
#include <io51.h>
#include <string.h>
#include <stdio.h>
#define config P1.0
```

```
void init();
void conf();
void conf_1();
void conf_2();
void conf_3();
void conf_4();
void conf_5();
void conf_6();
void menu();
void help();
void calc_time();
void set_time();
void set_date();
void send_data(char d[128]);
```

```
int sync=0,hour=0,min=0,sec=0,year=2004,month=1;
int day=1,leap_year=0,leap_sec=0;
int buffer=0,menu_flag=0,rs_data=0,conf_flag=0,i=0;
```

```
char time[7];
char date[9];
char enter[3]="\r\n";
char c_buf[7];
```



```

/*pääohjelma*/
main()
{
    init();

    for(;;)
    {
        if(sync==1)                /*uusi sekunnin pulssi*/
        {
            sync=0;                /*nollataan keskeytyslippu*/
            calc_time();            /*lisätään aikaa*/

            /*lähetetään minuutin välein, jos kytkin asennossa 0*/
            if(sec==0 && config==0)
            {
                REN=0;              /*sarjaportilla vastaanotto
                                   kielletty*/
                set_time();          /*ajan muuttaminen sopivaan
muotoon*/
                send_data(time);     /*ajan lähetys*/
                set_date();          /*päiväyksen muuttaminen sopivaan
                                   muotoon*/
                send_data(date);     /*päiväyksen lähetys*/
            }
            if(config==0)menu_flag=0;
        }

        /*tulostetaan menu, kun kytkin asennossa 1*/
        if(config==1 && menu_flag==0)
        {
            menu();                 /*tulostetaan menu*/
            menu_flag=1;            /*merkataan menu tulostetuksi*/
            REN=1;                  /*sallitaan sarjaportin
vastaanotto*/
        }

        /*kun sarjaporttiin tulee dataa ja kytkin on asennossa
1*/
        if(config==1 && rs_data==1)
        {
            rs_data=0;              /*nollataan data flaggi*/
            conf();                 /*tutkitaan mitä dataa tulee?*/
        }
    }
}

```

```

/*alustuksia*/
void init()
{
    /*keskeytykset*/
    TCON=0x01;          /*keskeytys 0 toimii laskevalla
reunalla*/
    IE=0x91;           /*sallitaan ulkoinen keskeytys 0 ja
                        sarjaportin keskeytys*/

    /*laskurin alustus*/
    PCON=0x00;          /*SMOD-bitti on 0*/
    TMOD=0x20;          /*laskuri 1, toimintamuoto 2*/
    TH1=0xFD;           /*nopeus 9600*/
    TR1=1;

    /*sarjaportin alustus*/
    SM0=0;              /*käytössä toimintamuoto 1*/
    SM1=1;              /*10-bitin kehys (start+8bit+stop)*/
    REN=0;              /*vastaanotto kielletty*/

    /*keskeytysten priorisointi*/
    IP=0x01;
}

/*sarjaportin keskeytys*/
interrupt void SCON_int(void)
{
    if(RI==1)
    {
        RI=0;
        TI=0;
        buffer=SBUF;    /*luettu merkki buffer muuttuun.
                        buffer tyyppiä int, koska ei
                        tarvitse lukea muita merkkejä. Merkkejä
                        käsitellään tästä eteenpäin
                        lukuina*/
        buffer=buffer-48; /*vähennetään 48, koska buffer on
ascii taulukon desimaalina*/
        rs_data=1;      /*merkataan, että tulee dataa*/
    }
}

```

```

/*ulkoinen keskeytys 0*/
interrupt void EX0_int()
{
    sync=1;                /*merkataan keskeytys tulleeeksi*/
}

```

```

/*datan lähetys*/
void send_data(char d[128])
{
    int i;
    for(i=0;i<strlen(d);i++)
    {
        TI=0;                /*TI-lipun nollaus*/
        SBUF=d[i];          /*lähetys*/
        while(TI==0);        /*odotetaan että TI-lippu asettuuu
ykköseksi*/
    }
}

```

```

/*käskyvalinta*/
void conf()
{
    if(conf_flag==2)          conf_2();
    if(conf_flag==4)          conf_4();
    if(conf_flag==6)          conf_6();

    if(buffer==1 && conf_flag==0) conf_1();
    if(buffer==2 && conf_flag==0)
    {conf_flag=2;send_data("\r\n>>");}
    if(buffer==3 && conf_flag==0) conf_3();
    if(buffer==4 && conf_flag==0)
    {conf_flag=4;send_data("\r\n>>");}
    if(buffer==5 && conf_flag==0) conf_5();
    if(buffer==6 && conf_flag==0)
    {conf_flag=6;send_data("\r\n>>");}
    if(buffer==24 || buffer == 15 || buffer == 56 &&
conf_flag==0)
        help();
}

```

```

/*ajan laskeminen*/
void calc_time()
{
    sec++;                /*lisätään sekunti*/

    /*karkaussekunti*/
    if      (leap_sec==1 && hour==23 && min==59 && sec==60 &&
             day==30 && month==6) {leap_sec=0;}
    else if (leap_sec==2 && hour==23 && min==59 && sec==60 &&
             day==31 && month==12){leap_sec=0;}
    else if (sec>59)      {sec=0;min++;}
    if (min==60)          {min=0;hour++;}
    if (hour==24)         {hour=0;min=0;sec=0;day++;}

    /*karkausvuosi, joka neljäs vuosi*/
    for(leap_year=year;leap_year>3;leap_year=leap_year-4){}

    /*päiväys*/
    /*tammikuu -> helmikuu*/
    if(month==1 && day==32){month=2; day=1;}
    /*helmikuu -> maaliskuu*/
    if(month==2 && day==29 && leap_year!=0){month=3; day=1;}
    /*helmikuu -> maaliskuu*/
    if(month==2 && day==30 && leap_year==0){month=3; day=1;}
    /*maaliskuu -> huhtikuu*/
    if(month==3 && day==32){month=4; day=1;}
    /*huhtikuu -> toukokuu*/
    if(month==4 && day==31){month=5; day=1;}
    /*toukokuu -> kesäkuu*/
    if(month==5 && day==32){month=6; day=1;}
    /*kesäkuu -> heinäkuu*/
    if(month==6 && day==31){month=7; day=1;}
    /*heinäkuu -> elokuu*/
    if(month==7 && day==32){month=8; day=1;}
    /*elokuu -> syyskuu*/
    if(month==8 && day==32){month=9; day=1;}
    /*syyskuu -> lokakuu*/
    if(month==9 && day==31){month=10;day=1;}
    /*lokakuu -> marraskuu*/
    if(month==10 && day==32){month=11;day=1;}
    /*marraskuu -> joulukuu*/
    if(month==11 && day==31){month=12;day=1;}
    /*joulukuu -> tammikuu*/
    if(month==12 && day==32){month=1; day=1;year++;}
}

```

```
/*ajan asettaminen char muuttuujaan*/
```

```
void set_time()
```

```
{
    if      (hour<10 && min<10 && sec<10)
        sprintf(time, "0%d:0%d:0%d",hour,min,sec);
    else if(hour<10 && min<10)
        sprintf(time, "0%d:0%d:%d" ,hour,min,sec);
    else if(hour<10 && sec<10)
        sprintf(time, "0%d:%d:0%d" ,hour,min,sec);
    else if(hour<10)
        sprintf(time, "0%d:%d:%d" ,hour,min,sec);
    else if(min<10 && sec<10)
        sprintf(time, "%d:0%d:0%d" ,hour,min,sec);
    else if(min<10)
        sprintf(time, "%d:0%d:%d" ,hour,min,sec);
    else if(sec<10)
        sprintf(time, "%d:%d:0%d" ,hour,min,sec);
    else
        sprintf(time, "%d:%d:%d" ,hour,min,sec);
}
```

```
/*päiväyksen asettaminen char muuttuujaan*/
```

```
void set_date()
```

```
{
    if      (day<10 && month<10)
        sprintf(date, "0%d.0%d.%d",day,month,year);
    else if(day<10)
        sprintf(date, "0%d.%d.%d" ,day,month,year);
    else if(month<10)
        sprintf(date, "%d.0%d.%d" ,day,month,year);
    else
        sprintf(date, "%d.%d.%d" ,day,month,year);
}
```

```
/*menu*/
```

```
void menu()
```

```
{
    send_data("\r\n>");
}
```



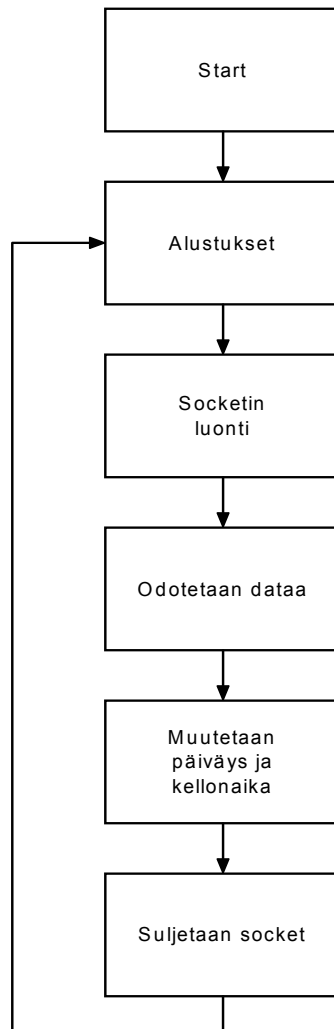


```
/*karkaussekuntin tarkastaminen*/
void conf_5()
{
    if(leap_sec==0){send_data("\r\nOFF");}
    if(leap_sec==1){send_data("\r\n30.6.");}
    if(leap_sec==2){send_data("\r\n31.12");}
    menu(); /*näytetään menu*/
}
```

```
/*karkaussekuntin asettaminen*/
void conf_6()
{
    leap_sec=buffer;
    conf_5();
    conf_flag=0;
    buffer=0;
}
```

```
void help()
{
    send_data("\r\n1. Lue kellonaika\r\n2. Aseta
kellonaika\r\n3.
        Lue päivämäärä");
    send_data("\r\n4. Aseta päivämäärä\r\n5. Tarkasta
        karkaussekunti\r\n6. Aseta karkaussekunti");
    menu();
}
```





```
#Jarno Hujanen
#Insinööriyö
#Kellonajanjakolaite
#Kajaanin ammattikorkeakoulu
```

```
#Ohjelma kuuntelee porttia 4000 (UDP) ja muuttaa
tietokoneen #kellonajan ja päivämäärän. Ohjelma toimii
ikuisessa #silmutuksessa ja käyttää WinSockia.
```

```
#include <stdio.h>
#include <winsock.h>
```

```
short port=4000;           //portti, jota
                           //kuunnellaan
```

```
void Server( ) ;
```

```
void main()
{
    WSADATA wsa;
    for(;;)                                //ikuinen silmukka
    {
        WSStartup(MAKEWORD(1,1), &wsa);    //käytetään WinSock
                                           //1.1:tä
        Server();                          //käynnistetään
serveri
        WSACleanup();                      //vapautetaan socket
    }
}
```

```

void Server()
{
    char buffer[256];                //data bufferin koko
    SOCKET s;
    s = socket(          AF_INET,          //ollaan Internetissä
                  SOCK_DGRAM,            //käytetään UDP:tä
                  IPPROTO_UDP);          //käytetään UDP:tä

    SOCKADDR_IN server;
    server.sin_family=AF_INET;
    server.sin_addr.s_addr=INADDR_ANY; //mistä tahansa
                                      //lähetetty
                                      //data
                                      //vastaanotetaan
    server.sin_port=htons(port);       //käytetään aiemmin
                                      //määriteltä porttia
bind(s,                             //sidotaan
    osoitetiedot
        (LPSOCKADDR)&server,
        sizeof(struct sockaddr)
    );

    memset(buffer, 0, sizeof(buffer)); //tyhjennetään bufferi
    recv(s,buffer,sizeof(buffer),0);   //odotetaan dataa

    if(strlen(buffer)!=0)               //jos bufferissa muuta
                                      //kuin tyhjää
    {
        {
            char set_data[32];
            if(buffer[2]==':')           //kellonaika
            {
                strcpy(set_data, "time "); //muuttuun teksti
time
                strcat(set_data, buffer); //lisätään kellonaika
                printf("\n%s", set_data);
                system(set_data);         //asetetaan kellonaika
                                      //dossin käskyllä:
                                      //time hh:mm:ss
            }
            if(buffer[2]=='.')           //päivämäärä
            {
                strcpy(set_data, "date ");
                strcat(set_data, buffer);
                printf("\n%s", set_data);
                system(set_data);
            }
        }
    }
    closesocket(s);                   //suljetaan socket

```

## Käyttöohje

Laite tarvitsee toimiakseen 8–15 V:n käyttöjännitteen, joka syötetään prosessorikortilla oleviin banaani liittimiin. Laitteessa olevalla kytkimellä valitaan, onko laite kytketty PC-tietokoneeseen vai sarjaliikennemuunttimeen. Kytkimen ollessa asennossa 1, voidaan laitteen asetuksia muuttaa tietokoneella sarjaportin kautta. Terminaali ohjelman asetukset täytyy olla 9600, N, 8, 1 ja vuonohjausta ei käytetä. Kellonajan jakolaitteen ja tietokoneen välillä käytetään ristiinkytettyä sarjakaapelia.

Laitteen ollessa alkutilassa näytöllä näkyy merkki >. Valikkorakenne on seuraavanlainen:

1. Lue kellonaika
2. Aseta kellonaika  
ttmmss
3. Lue päivämäärä
4. Aseta päivämäärä  
ppkkvvvv
5. Tarkasta karkaussekunti
6. Aseta karkaussekunti
  1. 30.6.
  2. 31.12.
- H. Tulosta menu
- ?. Tulosta menu

Kun valitaan 2, 4 tai 6 tulee näytölle lukemaan >>. Kun olet asettanut asetukset oikeiksi, irrota sarjakaapeli ja kytke kellonajan jakolaite sarjaliikennemuunttimeen, jonka jälkeen käännä kytkin asentoon 0.